UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/829,668	04/22/2004	Douglas C. Burger	UTOTC.000068	6831

42640 7590 08/26/2008

DILLON & YUDELL LLP
8911 NORTH CAPITAL OF TEXAS HWY
SUITE 2110
AUSTIN, TX 78759

EXAMINER	
FENNEMA, ROBERT E	
ART UNIT	**PAPER NUMBER**
2183	

MAIL DATE	DELIVERY MODE
08/26/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/829,668	BURGER ET AL.	
	Examiner	Art Unit	
	ROBERT E. FENNEMA	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 May 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *37-55* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *37-55* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 37-55 are pending. Claims 37-55 added as per Applicant's request. All prior claims cancelled as per Applicant's request.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 37-39, 43-48, and 50-55 are rejected under 35 U.S.C. 102(b) as being anticipated by Requa et al. ("The Piecewise Data Flow Architecture: Architectural Concepts", herein Requa).

4. As per Claim 37, Requa teaches: A method, comprising:

partitioning a program into a plurality of groups of instructions (Page 426, first column, second paragraph, instructions are grouped into blocks);

assigning a group of instructions selected from the plurality of groups of instructions to a plurality of interconnected preselected computation nodes (Page 426, first column, second paragraph, the blocks are sent to processors);

loading the group of instructions to the plurality of interconnected preselected computation nodes (Page 426, first column, second paragraph, also see Page 433, "The PDF Block Processor"); and

executing the group of instructions as each one of the instructions in the group of instructions receives all necessary associated operands for execution (Page 433, first column, third paragraph, an instruction waits for input operands, then is executed).

5. As per Claim 38, Requa teaches: The method of claim 37, wherein at least one computation node included in the plurality of interconnected preselected computation nodes has at least one input port capable of being coupled to at least one preselected first other computation node included in the plurality of interconnected preselected computation nodes (Figure 1),

the input port to receive input data (Page 427, second column, Paragraph 2 (any consumer can receive any data), also see Page 433, The PDF Block Processor (herein referred to as PDF for the remainder of this claim, as this section will be referenced several times). Additionally, one would recognize a processor/execution unit inherently requires an input port to receive data),

a first store coupled to the at least one input port to store the input data (PDF, paragraph 1, input operands are stored in registers),

a second store coupled to an instruction sequencer, the second store to receive and store the at least one instruction (PDF, Paragraph 1, the instruction issue section holds instructions prior to execution),

an instruction wakeup unit to match the input data to the at least one instruction (PDF, paragraph 1, the operand source fields are modified as data comes in), at least

one execution unit to execute the at least one instruction using the input data to produce output data (PDF, paragraph 1, the instructions are executed after receiving inputs),

at least one output port capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes (Figure 1 and Page 427, see below), and

a router to direct the output data from the at least one output port to the at least one preselected second other computation node (Page 427, second column, paragraph 2. Any consumer can receive any data from the interconnection network, thus all processors are capable of sending data to any other processor).

6. As per Claim 39, Requa teaches: The method of claim 37, wherein at least one of the plurality of groups of instructions is a basic block (Page 426, first column third paragraph).

7. As per Claim 43, Requa teaches: The method of claim 37, wherein loading the group of instructions to the plurality of interconnected preselected computation nodes includes:

sending at least two instructions selected from the group of instructions from an instruction sequencer to a selected computation node included in the plurality of interconnected preselected computation nodes for storage in a store (Page 426, Column 1, Paragraph 2, each instruction is sent out to a selected processor/functional

unit/node).

8. As per Claim 44, Requa teaches: The method of claim 37, wherein executing the group of instructions as each one of the instructions in the group of instructions receives all necessary associated operands for execution includes:

matching at least one instruction selected from the group of instructions with at least one operand received from an other computation node included in the plurality of interconnected preselected computation nodes (Page 427, second column, second paragraph, where any consumer (processor) can receive any data, and instructions waiting to execute wait for results from the processor before executing).

9. As per Claim 45, Requa teaches: The method of claim 37, wherein loading the group of instructions to the plurality of interconnected preselected computation nodes includes:

Sending a first set of instructions selected from a first group of instructions selected from the plurality of groups of instructions from an instruction sequencer to the plurality of interconnected preselected computation nodes for storage in a first frame included in a first computation node included in the plurality of interconnected preselected computation nodes (Page 433, PDF, paragraph 1, the instruction issue section is a buffer where instructions wait to be executed); and

sending a second set of instructions selected from the first group of instructions from the instruction sequencer to the plurality of interconnected preselected

computation nodes for storage in a second frame included in the first computation node (Page 433, PDF, paragraph 1, the instruction issue section is a buffer where instructions wait to be executed).

10. As per Claim 46, Requa teaches: The method of claim 37, wherein assigning a group of instructions selected from the plurality of groups of instructions to a plurality of interconnected preselected computation nodes includes:

assigning a first group of instructions to a first set of frames included in the plurality of interconnected preselected computation nodes (Page 433, PDF, paragraph 1, the instruction issue section is a buffer where instructions wait to be executed);

assigning a second group of instructions to a second set of frames included in the plurality of interconnected preselected computation nodes (Page 433, PDF, paragraph 1, the instruction issue section is a buffer where instructions wait to be executed), wherein the first group and the second group of instructions are capable of concurrent execution (Abstract), and wherein at least one output datum associated with the first group of instructions is written to a register file and passed directly to the second group of instructions for use as an input datum by the second group of instructions (Page 430, first column, second paragraph, when an instruction completes, the results are written to a register, which is then read by the dependent instruction).

11. As per Claim 47, Requa teaches: An article comprising a machine-accessible medium having associated data, when accessed, results in a machine performing:

loading a group of instructions to a plurality of interconnected preselected computation nodes (Page 426, first column, second paragraph, also see Page 433, "The PDF Block Processor"),

wherein a program is partitioned into a plurality of groups of instructions (Page 426, first column, second paragraph, instructions are grouped into blocks),

wherein the group of instructions from a plurality of groups of instructions is assigned to the plurality of interconnected preselected computation nodes (Page 426, first column, second paragraph, the blocks are sent to processors); and

executing the group of instructions as each one of the instructions in the group of instructions receives all necessary associated operands for execution (Page 433, first column, third paragraph, an instruction waits for input operands, then is executed).

12. As per Claim 48, Requa teaches: The article of claim 47, wherein partitioning the program into the plurality of groups of instructions is performed by a compiler (Page 429, first column, "PDF Architecture").

13. As per Claim 50, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes information, which when accessed by the machine, results in the machine performing:

statically assigning all of the plurality of groups of instructions for execution (Page 432, see Figure 8, and column 1, paragraph 2).

14. As per Claim 51, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes information, which when accessed by the machine, results in the machine performing:

dynamically issuing one or more instructions from at least one of the plurality of groups of instructions for execution (Page 426, second column, third paragraph. Instructions are issued as dependent operands come in).

15. As per Claim 52, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes information, which when accessed by the machine, results in the machine performing:

generating a wakeup token to reserve an output data channel to connect selected computation nodes included in the plurality of interconnected preselected computation nodes (Page 427, section column, second paragraph).

16. As per Claim 53, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes data, which when accessed by the machine, results in the machine performing:

detecting execution termination of the group of instructions including an output having architecturally visible data (Page 433, second column first paragraph, a flag is set when an execution is done, also see Page 430, second paragraph); and

committing the architecturally visible data to a register file (Page 430, second paragraph).

17. As per Claim 54, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes information, which when accessed by the machine, results in the machine performing:

detecting execution termination of the group of instructions including an output having architecturally visible data (Page 433, second column first paragraph, a flag is set when an execution is done, also see Page 430, second paragraph); and

committing the architecturally visible data to a memory (Page 430, second paragraph).

18. As per Claim 55, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes data, which when accessed by the machine, results in the machine performing:

routing an output datum arising from executing the group of instructions to a consumer node included in the plurality of interconnected preselected computation nodes, wherein the address of the consumer node is included in a token associated with at least one instruction included in the group of instructions (Page 429, second column, second paragraph).

Claim Rejections - 35 USC § 103

19. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

20. Claims 40-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Requa, in view of Official Notice.

21. As per Claim 40, Requa teaches: The method of claim 37, but fails to teach: wherein at least one of the plurality of groups of instructions is a hyperblock.

Requa teaches of a system which uses basic blocks, but does not teach that the groups may be hyperblocks. However, Examiner is taking Official Notice that one of ordinary skill in the art would be capable and motivated to use hyperblocks in lieu of basic blocks, to take advantage of the ability to have multiple exits from a block (While Applicant has not provided a definition of a hyperblock, Examiner has found it to represent a block with one entrance and potentially (but not necessarily) more than one exit).

22. As per Claim 41, Requa teaches: The method of claim 37, but fails to teach: wherein at least one of the plurality of groups of instructions is a superblock.

Requa teaches of a system which uses basic blocks, but does not teach that the groups may be superblocks. However, Examiner is taking Official Notice that one of

ordinary skill in the art would be capable and motivated to use superblocks in lieu of basic blocks, to take advantage of the ability to have multiple exits from a block (While Applicant has not provided a definition of a superblock, Examiner has found it to represent a block with one entrance and potentially (but not necessarily) more than one exit, however, Examiner is unclear how a superblock is different from a hyperblock).

23. Claims 42 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Requa, in view of Fisher.

24. As per Claim 42, Requa teaches: The method of claim 37, but fails to teach:

wherein at least one of the plurality of groups of instructions is an instruction trace constructed by a hardware trace construction unit at run time.

While Requa teaches the method as disclosed in Claim 37, Requa does not teach about traces, or a trace construction unit to construct such a trace. However, Fisher teaches of Trace Scheduling, where the basic blocks used by Requa are compacted, and instead use traces (Page 462, Section D). The advantage to this compaction method using traces allows for more efficient parallel code, done in a manner far more efficient than previous methods (Abstract). Given this advantage, one of ordinary skill in the art would have been motivated to use these traces in place of the basic blocks as taught by Requa to further increase the efficiency of the system.

25. As per Claim 49, Requa teaches: The article of claim 47, but fails to teach:

wherein partitioning the program into the plurality of groups of instructions is performed by a run-time trace mapper.

While Requa teaches the article as disclosed in Claim 47, Requa does not teach that the partitioning of the program is done by a trace mapper. However, Fisher teaches of Trace Scheduling, where the basic blocks used by Requa are compacted, and instead use traces (Page 462, Section D), created and optimized by a scheduler (Page 482, second column, second paragraph). The advantage to this compaction method using traces allows for more efficient parallel code, done in a manner far more efficient than previous methods (Abstract). Given this advantage, one of ordinary skill in the art would have been motivated to use these traces in place of the basic blocks as taught by Requa to further increase the efficiency of the system.

Response to Arguments

26. Applicant has argued that Requa does not teach assigning a group of instructions selected from the plurality of groups of instructions to a plurality of interconnected preselected computation nodes, by arguing that instead, Requa discloses that instructions are grouped into blocks, and that when a block executes, each instruction is scheduled for execution on a processor. This is clearly what the claim language requires, a group of instructions (the block) is selected among a plurality of groups (multiple blocks), and assigned to a plurality of computation nodes (Page 426, Requa teaches that each instruction is assigned to one of the many nodes (processors)), thus the group of instructions is assigned to a plurality of nodes. Both the

claim language, and the teachings of Requa are quite plain, and Requa maps to the limitation, as Requa teaches a group of instructions assigned to a plurality of computation nodes.

Applicant has further argued that Requa does not teach the "loading" limitation, by claiming that Requa loads the block from memory, schedules an instruction on a processor, stores the data in registers, modifies the source operands for each instruction, and then executes them when it is ready. Examiner asserts that this is also what has been claimed. Requa loads the block from memory, puts the instructions into registers in the processors, and executes them. Again, the plain language of the claim is quite clear, and maps to Requa, and Applicant's arguments appear to support the Examiners rejection.

Applicant further argues Claim 38, arguing that Requa does not teach the limitations of the computation node, however, Examiner has clearly laid out the parts of Requa that teach these limitations, while Applicant has made arguments that Requa does not teach these features, and has stated what they believe Requa teaches, the Applicant has not provided any kind of argument as to why the Examiners interpretation of Requa does not fit the claim language, and instead has just broadly dismissed the Examiners mapping to the claims, and as a result, the Examiner is not persuaded that the mapping is incorrect, as the Examiner has clearly pointed out language which does teach these features.

Applicant further argues on Page 11 that Requa does not teach sending at least two instructions from a group of instructions to a selected computation node, however,

Examiner notes that on Page 426 of Requa, that a block of instructions (the group) contains up to 255 instructions, which is clearly at least 2 instructions. And the instruction sequencer Applicant argues Requa does not have is the unit which sends the block to the processor. Again, Applicant has broadly argued that Requa does not teach a claim limitation, instead of pointing out how the Examiners mapping fails to teach the limitation.

On Page 12, Applicant argues that Requa does not teach "matching at least one instruction selected from the group of instructions with at least one operand received from an other computation node included in the plurality of nodes, and again, does not explain why the Examiners interpretation and mapping of the claim is incorrect, but again just states what Requa does, and broadly claims that it is not taught. Requa teaches that instructions must wait for their data operands before being able to proceed, and teaches that every processor is connected to every other processor, so the producing processors can send data to the consumer instructions (Page 427), thus satisfying the claim language.

On Pages 12 and 13, Applicant again argues that Requa does not teach a claim limitation, and states what they believe Requa teaches, but again, does not explain why the Examiners mapping of the claim language is incorrect, and instead just states that "there is no language in the cited passage that discloses the limitation". Examiner refers to the rejection, where he has mapped out clearly how Requa reads on the claim, and seeing no argument as to why this is incorrect, Examiner is not persuaded that there is

no language in the passage teaching the limitation, as the cited portion teaches the limitation.

Applicants arguments on Pages 14-16 appear to be similar to the arguments made in a similar claim about frames. However, again, Applicant does not directly argue the Examiners rejection, and instead, broadly states that there is no language in the cited passages to teach the limitation, which Examiner does not find persuasive, and refers to his above rejection.

Regarding Applicants arguments on Page 16, Examiner notes in Figure 8 that the blocks of instructions must be executed statically, and that each block must wait until the previous one is complete, meaning that all instructions in a block are scheduled for execution, a static determination, since they are all done. Therefore no extrinsic evidence is required as this limitation is clearly taught by Requa.

Regarding the remaining arguments made under 102, Applicant again has just stated that every single limitation claimed is not taught by Requa, without making any specific remarks about why the Examiners mapping is incorrect or invalid, therefore, Examiner is not persuaded by the Applicants assertion that it is.

Applicant has further argued, on Pages 19 and 20, that Examiner has not provided “articulated reasoning with some rational underpinning to support the legal conclusion of obviousness”. However, Examiner believes he has done so, as laid out in the rejection, thus Examiner finds these arguments non-persuasive. Examiner has clearly laid out why one of ordinary skill in the art would see the advantages in using a

superblock or hyperblock, and as these are terms known in the art, and defined in Fisher, with advantages laid out as well, one of ordinary skill in the art would be capable and motivated to modify Requa in order to take advantages of well known and documented advantages from making use of either a hyperblock or superblock.

Conclusion

27. This is an request for continued examination of the current Applicant. All claims are drawn to the same invention claimed in the earlier application and could have been finally rejected on the grounds and art of record in the next Office action if they had been entered in the earlier application. Accordingly, **THIS ACTION IS MADE FINAL** even though it is a first action in this case. See MPEP § 706.07(b). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no, however, event will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ROBERT E. FENNEMA whose telephone number is (571)272-2748. The examiner can normally be reached on Monday-Friday, 8:30-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

Robert E Fennema
Examiner
Art Unit 2183

RF